The k-BDH Assumption Family: Bilinear Map Cryptography from Progressively Weaker Assumptions

Karyn Benson[1], Hovav Shacham[1,⋆], and Brent Waters[2,⋆⋆]

[1] University of California, San Diego
{kbenson,hovav}@cs.ucsd.edu
[2] University of Texas at Austin
bwaters@cs.utexas.edu

Abstract. Over the past decade bilinear maps have been used to build a large variety of cryptosystems. In addition to new functionality, we have concurrently seen the emergence of many strong assumptions. In this work, we explore how to build bilinear map cryptosystems under progressively weaker assumptions.

We propose k-BDH, a new family of progressively weaker assumptions that generalizes the decisional bilinear Diffie-Hellman (DBDH) assumption. We give evidence in the generic group model that each assumption in our family is strictly weaker than the assumptions before it. DBDH has been used for proving many schemes secure, notably identity-based and functional encryption schemes; we expect that our k-BDH will lead to generalizations of many such schemes.

To illustrate the usefulness of our k-BDH family, we construct a family of selectively secure Identity-Based Encryption (IBE) systems based on it. Our system can be viewed as a generalization of the Boneh-Boyen IBE, however, the construction and proof require new ideas to fit the family. Our methods can be extended to produce hierarchical IBEs and CCA security; and give a fully secure variant. In addition, we discuss the opportunities and challenges of building new systems under our weaker assumption family.

1 Introduction

Since the introduction of the Boneh-Franklin [1] Identity-Based Encryption (IBE) system a decade ago, we have seen an explosion of new cryptosystems based on bilinear maps. These systems have provided a wide range of functionality including: new signature systems, functional encryption, e-cash, "slightly" homomorphic encryption, broadcast encryption and oblivious transfer to name just

⋆ Supported by the MURI program under AFOSR Grant No. FA9550-08-1-0352.
⋆⋆ Supported by NSF CNS-0915361 and CNS-0952692, AFOSR Grant No: FA9550-08-1-0352, DARPA PROCEED, DARPA N11AP20006, Google Faculty Research award, the Alfred P. Sloan Fellowship, and Microsoft Faculty Fellowship.

E. Dawson (Ed.): RSA 2013, LNCS 7779, pp. 310–325, 2013.

a few. The focus of many of this work was to develop new (and often not realized before) functionality. While Boneh-Franklin and many first IBE systems used "core" assumptions such as the Bilinear Diffie-Hellman or decisional variants, over time there has been a trend in bilinear map based work to employ stronger assumptions in order to obtain these functionalities. Examples of these assumptions range from "q-type" [2] assumptions, assumptions in composite order groups [3], interactive assumptions [4] and proofs that appealed directly on the generic group model [5,6]

Interestingly, even some work that focused on tightening security (versus achieving new functionality) have had to employ relatively strong assumptions. For example, Gentry and Halevi [7] and Waters [8] proposed two different approaches for solving the problem of achieving adaptive security for Hierarchical Identity-Based encryption. To achieve this the former used a q-type assumption where the strength of the assumption depends on the number of attacker private key queries. The latter used the decisional-Linear assumption, where the target of the assumption is in the source element of the bilinear group versus the target element. Both of these assumptions are potentially stronger than the classic decisional-BDH prior IBE and related systems were built upon.

Our Goals. In this work, we move in the opposite direction of this trend. We will build bilinear map systems that depend on *weaker* assumptions than the decisional-BDH assumption. In particular, we want to create a suitable family of assumptions that becomes progressively weaker as some parameter k is increased. Therefore one can increase k as a hedge against potential future attacks such as an n-linear map for $n > 2$.

A natural starting point for our investigation is the k-Linear family of assumptions [9,10], which generalizes the decisional Diffie-Hellman assumption (DDH) and the decisional Linear assumption of Boneh, Boyen, and Shacham [11]. For $k \geq 1$, a k-Linear problem instance is a tuple $(g, g_1, \ldots, g_k, g_1^{r_1}, \ldots, g_k^{r_k}, T)$, where the generators are random in the group $\mathbb{G}$, the exponents in its support $\mathbb{Z}_p$, and the goal is to determine whether T is equal to $g^{r_1+\cdots+r_k}$ or random. DDH is 1-Linear, and the Linear assumption is 2-Linear.

The k-Linear assumption family has been successfully used to build chosen ciphertext secure encryption [9,10]; to construct pseudorandom functions [12,13]; to construct public-key encryption secure in the presence of encryption cycles [14,15] and public-key encryption resilient to key leakage [16,17]; to construct lossy trapdoor functions [18]; to construct leakage-resilient signatures [19].

While the k-Linear family has been successful in the above contexts, we desire an assumption that can be used in bilinear map cryptosystems in place of where DBDH has typically been applied. Here using the k-Linear family does not appear well suited for two reasons. First, since the assumption of the family operates solely in the source group, the assumption is not even "aware" of bilinear groups. Therefore it is not clear how it might be applied in certain systems (e.g. an variant of Boneh-Boyen IBE) where we are hiding a message in the target group. Second, the Linear assumption family has an inconsistent interaction with the DBDH assumption: the $1, 2$-Linear assumptions are actually stronger

than DBDH, but the the k-Linear assumptions for $k > 2$ are generically incomparable to DBDH. One reason that the (2-)Linear assumption has proved so useful is that it gives DBDH "for free," but this is lost as soon as one increases k beyond 2. If a new IBE system were based on k-Linear *and* DBDH, it is not clear that this would provide an improvement in security.[1]

Our goals, then, are to find an assumption family that meets the following criteria:

- As we increase the assumption family parameter k, we should become more confident in the security of our assumption. In particular, we would argue that our k parameterized assumption is in some sense more secure than both existing decisional assumptions in bilinear groups and more secure than the $k-1$ instance.
- Our family of assumptions should be amenable to building cryptographic systems. Ideally, for any system built using the DBDH assumption, one could find a variant built using our family.

The k-BDH Family of Assumptions. Our main contribution is a new family of assumptions that can serve as a weaker generalization of DBDH.

We propose a family of progressively weaker assumptions, the k-BDH assumptions, that generalizes the DBDH assumption. The 1-BDH assumption is equivalent to DBDH. More generally, the k-BDH assumption is as follows:

$$\text{given } g,\ g^x, g^y,\ v_1, \ldots, v_k,\ v_1^{r_1}, \ldots, v_k^{r_k} \text{ in } \mathbb{G},$$
$$\text{decide whether } T = e(g,g)^{(xy)(r_1+\cdots+r_k)} \text{ or random in } \mathbb{G}_T.$$

Here g and $\{v_i\}$ are random generators of $\mathbb{G}$ and x, y, and $\{r_i\}$ are random elements of its support $\mathbb{Z}_p$. We consider only the decisional versions of these problems; as with k-Linear, the computational versions are all equivalent to each other. (This is also why we refer to our assumption family as k-BDH and not k-DBDH; there is no interesting family of computational assumptions from which our decisional assumptions must be distinguished.)

We remark that discovering and choosing such a family turned out to be challenging. Initially, we considered the assumption family in which the adversary, given the same input values in $\mathbb{G}$, must distinguish $\prod_i e(g, v_i)^{xyr_i}$ from random in $\mathbb{G}_T$. This assumption family is easier to use than our k-BDH because the values v_i and $v_i^{r_i}$ are available to pair with g^x or g^y, the way that in DBDH we can use the pairing to compute any of $e(g,g)^{xy}$, $e(g,g)^{xz}$, $e(g,g)^{yz}$. However, it turns out that every member of this alternative assumption family is equivalent to DBDH.[2] The fact that the values $\{g^{r_i}\}$ are not supplied in the k-BDH challenge make constructing an IBE from k-BDH more challenging.

[1] Similarly, for attribute-based encryption, if attribute-hiding were established based on k-Linear, but payload-hiding were established based on DBDH, then one the assumption for one property would be weakened while the assumption for the other property would remain strong.

[2] The reduction makes use of the DBDH tuple $(g, \prod_i v_i^{r_i}, g^x, g^y, C \stackrel{?}{=} \prod_i e(g, v_i)^{xyr_i})$.

We justify our choice by arguing both that the k-BDH assumptions are no stronger than existing (decisional) assumptions in bilinear groups and that it is plausible that they are strictly weaker. The former follows in a relatively straightforward by finding appropriate reductions. We can show that in a given group the k-BDH assumption is no stronger than DBDH and for a given k the k-BDH assumption is no stronger than the k-Linear assumption, for all values of k.

Arguing that the assumptions are weaker is more nuanced. Whether certain assumptions hold or do not hold might vary with the choice of a group and clearly if $\mathcal{P} = \mathcal{NP}$ all assumptions are equally false. We give evidence that, for each k, the $(k+1)$-BDH assumption is strictly weaker than the k-BDH assumption (i.e., the $(k+1)$-BDH problem is strictly harder to solve than the k-BDH problem). As in previous proofs of this sort for Linear [11] and k-Linear [9], we rely on an argument in the generic group model [20,21]. We show that the $k+1$-BDH problem is generically hard even in the presence of an oracle that solves k-BDH.

We demonstrate the utility of our assumption family, by constructing a family of IBEs secure under k-BDH. The size of the public parameters, secret keys, and ciphertexts are all linear in the parameter k. One can view our family as a generalization of the Boneh-Boyen selectively secure IBE system [22].

In the full version of the paper [23], we extend our construction family to a family of hierarchical IBEs. These yield CCA-secure schemes via standard transformations [24,25]. In addition, we show how to produce a Waters-IBE–style variant [26] that is fully secure in the standard model.

Looking Ahead. In the future, we expect that one will be able to build cryptosystems from our k-BDH assumption where DBDH was previously used. However, as our experience with IBE has taught us, this might require new insights or techniques.

One interesting challenge is whether one can build more complex systems using the k-BDH assumption where the performance overhead is *additive* in k versus a multiplicative factor (which seems more natural). For instance, in existing (Key-Policy) Attribute-Based Encryption [27,28] systems, the size of a private key is proportional to a policy expressed as a boolean formula. If, the cost of using the k-BDH assumption only required adding $\approx k$ more group elements, this could be a relatively small key size overhead for reasonably chosen k. This is in contrast to blowing up the entire key size by a factor of k. A similar argument holds for other parameters such as ciphertext size and decryption time. In one datapoint suggesting that this might be possible, Freeman et. al. [18] recently built Lossy Trapdoor Functions in a novel way from the k-linear assumption which were rather efficient relative to the "natural" extension of the Peikert and Waters [29] DDH construction.

There also exist currently exist several functionalities where there are no known systems that reduce to DBDH. These include systems that appear to inherently on assumption related to source group elements such as Decision Linear. Examples of these include Groth-Sahai NIZKs [30], dual system encryption proofs [8], and the Boneh-Goh-Nissim [3] slightly 2-homomorphic encryption system.

Finally, an interesting question is which k values one might use in practice. For very large k, it might turn out that bit by bit encryption systems built from using hard core bits [31] and Computational Diffie-Hellman or Computational Bilinear Diffie-Hellman have comparable efficiency. When proposing systems, it is important to keep in mind where these lines cross. However, we believe for most practical choices of k the k-BDH assumption will yield more efficient systems.

2 The *k*-BDH Assumption and Relationships

Throughout this paper we work in a cyclic group $\mathbb{G}$ of order p where p is a large prime. g is a generator of $\mathbb{G}$. $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ denotes an admissible bilinear map where $\mathbb{G}_T$ is another cyclic group of order p. The standard definitions of bilinear maps and well known complexity assumptions BDH, DBDH, Linear [11] and k-Linear [10] are used.

Definition 1. *The k-BDH problem in $\langle \mathbb{G}, \mathbb{G}_T, e \rangle$ asks given $(g, g^x, g^y, v_1, \ldots, v_k, v_1^{r_1}, \ldots, v_k^{r_k}, T)$ for $x, y, r_1, \ldots, r_k, c \in \mathbb{Z}_p^*$, $g, v_1, \ldots, v_k \in \mathbb{G}$ and $T \in \mathbb{G}_T$ does $T = e(g,g)^{xy(r_1+\cdots+r_k)}$ or is it the case that $T = e(g,g)^c$. An adversary, $\mathcal{B}$ outputs 1 if $T = e(g,g)^{xy(r_1+\cdots+r_k)}$ and 0 otherwise. $\mathcal{B}$ has advantage ϵ in solving k-BDH if*

$$|Pr[\mathcal{B}(g, g^x, g^y, v_1, \ldots, v_k, v_1^{r_1}, \ldots, v_k^{r_k}, e(g,g)^{xy(r_1+\cdots+r_k)}) = 1] - Pr[\mathcal{B}((g, g^x, g^y, v_1, \ldots, v_k, v_1^{r_1}, \ldots, v_k^{r_k}, e(g,g)^c) = 1]| \geq 2\epsilon.$$

Where the probability is taken over the random choice of $x, y, r_1, \ldots, r_k, c \in \mathbb{Z}_p^*$, $g, v_1, \ldots, v_k \in \mathbb{G}$ and the random bits consumed by $\mathcal{B}$.

The k-BDH Assumption is that if no t-time algorithm can achieve advantage at least ϵ in deciding the k-BDH problem in $\mathbb{G}$ and $\mathbb{G}_T$.

This is only a decisional problem. We show that, as a corollary of Theorem 4, the computational version is equivalent to the computational BDH problem.

2.1 *k*-BDH's Relationship to Standard Assumptions

In this subsection we state k-BDH's relationship to standard cryptographic assumptions; the proofs are straightforward and given in the full version [23].We also note that k-BDH is a member of the (R,S,T,f)-Diffie Hellman uber- assumption family [6]. Namely: $R = S = \{1, x, y, a_1, \ldots, a_k, a_1 r_1, \ldots, a_k r_k\}$, $T = \{1\}$ and $f = xy(r_1 + \cdots + r_k)$ where $v_i = g^{a_i}$ for $1 \leq i \leq k$. Being part of this family tells us that it is generically secure, however, the focus on our work is to understand the relative strengths of assumptions (discussed in Section 4).

***k*-BDH's Relationship to *k*-Linear.** We will use the notation L_k to denote the k-Linear problem. If we wish to specify the k-Linear assumption in a specific group $\mathbb{G}$ we write $L_k^{\mathbb{G}}$, and similarly for $\mathbb{G}_T$.

Theorem 1. *If the $L_k^{\mathbb{G}}$ assumption holds, then so does the k-BDH assumption.*

Theorem 2. *If the k-BDH assumption holds, then so does the $L_k^{\mathbb{G}_T}$ assumption.*

Evidence that k-BDH is not equivalent to either $L_k^{\mathbb{G}}$ or $L_k^{\mathbb{G}_T}$. From the above theorems, the natural question arises: Is k-BDH equivalent to the linear assumption in either $\mathbb{G}$ or $\mathbb{G}_T$? Such an equivalence would imply that k-BDH assumption is neither a new assumption nor a new tool to construct a family of IBEs. Fortunately, separation of the assumptions appears to be related to the hard problem of inverting a bilinear map [32,33]. We show separation results for these assumptions in the full version of this paper [23] in the generic group model.

k-BDH's Relationship to BDH

Theorem 3. *If the DBDH assumption holds, then so does the k-BDH assumption.*

Theorem 4. *If the Computational k-BDH assumption holds, then so does the Computational BDH assumption.*

Corollary 1. *The Computational k-BDH assumption is equivalent to the BDH assumption.*

Corollary 2. *The DBDH assumption is equivalent to the 1-BDH assumption.*

3 A Selectively Secure IBE System from the *k*-BDH Assumption

The standard definitions of IBE [1] and the selective-ID model [34] are used.

Using the k-BDH assumption in to create an IBE system is not straightforward. The main technical difficulty arises because the target in the k-BDH assumption, $(e(g,g)^{xy\sum_{i=1}^{k} r_i})$, is naturally an embedding of k Computational BDH problems: Given (g, g^x, g^y, g^{r_i}) find $e(g,g)^{xyr_i}$. However, we do not have the value g^{r_i} for each i. Instead, we have the pair $(v_i, v_i^{r_i})$, where v_i is a generator not used elsewhere.

We use a cancellation trick to effectively switch the base of the $v_i^{r_i}$. The setup algorithm will provide the values $e(g^x, v_i^{r_i})$ and v_i which are both taken to the same power in the encryption algorithm, namely y_i. The challenge needs to be crafted so that it takes $e(g^x, g^{r_i})$ to the power y instead of taking $e(g^x, v_i^{r_i})$ to the power y_i. To do this, we provide g^y in place of $v_i^{y_i}$. Since $v_i = g^{s_i}$ for some value of s_i we implicitly set $y_i = y/s_i$. Using the bilinear property of e, this effectively changes the value of the other term to $e(g^x, v_i^{r_i})^{y_i} = e(g,g)^{xr_is_i\frac{y}{s_i}} = e(g^x, g^{r_i})^y$. The product of these values is exactly the target of the k-BDH assumption.

To ensure the challenge has the proper distribution in the view of the adversary it is required to randomize g^y for every value of k.

Our IBE construction is related to the Boneh-Boyen scheme in the selective-ID model [22], which is proven secure under the DBDH assumption. To prove our scheme is secure under the k-BDH assumption requires an alteration to the "Boneh-Boyen trick" for generating the private key for identities other than the target identity.

The "Boneh-Boyen trick" raises elements of the DBDH instance to cleverly selected random values to obtain a valid private key. However, constructing the same private key with the $\mathsf{KeyGen(ID)}$ algorithm is impossible as the random selections are unknown. Our construction uses the same idea but using multiple bases $(g,\ v_i)$ requires *three* components instead of two for the first term of the private key.

Specifically, we use $(v_i^{\hat{r_i}})^{-t_i/d} v_i^{t_i m_i} (g^x)^{dm_i}$ for the first term. $v_i^{\hat{r_i}}$ is the randomization of $v_i^{r_i}$ that permits the challenge have the proper distribution. The value d is a function of the target identity and the identity associated with the private key; t_i is used to randomize a public parameter; and m_i randomizes the private key. The first term is dependent on both g^x and $v_i^{r_i}$ from the k-BDH assumption.

The IBE system works as follows:

Setup : The public parameters are $(g, u = g^x, v_1 = g^{s_1}, \ldots, v_k = g^{s_k}, v_1^{\hat{r_1}}, \ldots, v_k^{\hat{r_k}}, w_1, \ldots, w_k)$. The values $s_1, \ldots, s_k, \hat{r_1}, \ldots, \hat{r_k}, x$ (chosen uniformly and independently at random) are kept as the master-key.

$\mathsf{KeyGen(ID)}$: Select random $n_1, \ldots, n_k \in \mathbb{Z}_p^*$. For each $1 \leq i \leq k$ output $(K_{A,i}, K_{B,i})=((g^{x\hat{r_i}}(w_i u^{\mathsf{ID}})^{n_i},\ v_i^{n_i})$.

$\mathsf{Encrypt}(m, \mathsf{ID})$: Select random $y_1, \ldots, y_k \in \mathbb{Z}_p^*$. Output $C_0 = m \prod_{1 \leq i \leq k} e(g^x, v_i^{\hat{r_i}})^{y_i}$ and for each $1 \leq i \leq k$ output $(C_{A,i}, C_{B,i})=(v_i^{y_i}, (w_i u^{\mathsf{ID}})^{y_i})$ for a total of $2k+1$ values.

$\mathsf{Decrypt}(c)$: Output

$$\frac{C_0 \cdot \prod_{1 \leq i \leq k} e(K_{B,i}, C_{B,i})}{\prod_{1 \leq i \leq k} e(K_{A,i}, C_{A,i})} = \frac{m \prod_{1 \leq i \leq k} e(g^x, v_i^{\hat{r_i}})^{y_i} \cdot \prod_{1 \leq i \leq k} e(v_i^{n_i}, (w_i u^{\mathsf{ID}})^{y_i})}{\prod_{1 \leq i \leq k} e(g^{x\hat{r_i}}(w_i u^{\mathsf{ID}})^{n_i}, v_i^{y_i})} = m$$

3.1 Proof of Security

Theorem 5. *Suppose the k-BDH assumption holds in $\mathbb{G}$ and $\mathbb{G}_T$ (precisely, no t-time algorithm has advantage at least ϵ in solving the k-BDH problem in $\mathbb{G}$ and $\mathbb{G}_T$). Then the previously defined IBE system is $(t - \Theta(\tau kq), q, \epsilon)$-Selective-ID IND-CPA secure where τ is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the IBE system. We build algorithm $\mathcal{B}$ to solve a decisional k-BDH instance $(g, g^x, g^y, v_1, \ldots, v_k, v_1^{r_1}, \ldots, v_k^{r_k}, T \stackrel{?}{=} e(g,g)^{xy(r_1+\cdots+r_k)})$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows: Init: The selective identity games begins with $\mathcal{A}$ outputting an identity to attacked ID^*.

Setup: Algorithm $\mathcal{B}$ first selects random a_i, t_i for $1 \le i \le k$. It then sets the public parameters to: $(g, u = g^x, v_1, \ldots, v_k, v_1^{\hat{r_1}} = (v_1^{r_1})^{1/a_1}, \ldots, v_k^{\hat{r_k}} = (v_k^{r_k})^{1/a_k}, w_1 = v_1^{t_1}(g^x)^{-\mathsf{ID}^*}, \ldots, w_k = v_k^{t_k}(g^x)^{-\mathsf{ID}^*})$. These parameters are are all independent of ID^* in the view of $\mathcal{A}$. The a_i terms will serve as the way to randomize the challenge.

Phase 1: $\mathcal{A}$ issues queries for the private key of an identity, ID. It must be the case that $\mathsf{ID} \neq \mathsf{ID}^*$. $\mathcal{B}$'s response is generated as follows for each value of $1 \le i \le k$:

Select random m_i. Let $d = \mathsf{ID} - \mathsf{ID}^*$. Output $(K_{A,i}, = (v_i^{\hat{r_i}})^{-t_i/d} v_i^{t_i m_i}(g^x)^{dm_i}$, $K_{B,i} = (v_i^{\hat{r_i}})^{(-1/d)} v_i^{m_i}$. For $n_i = -\hat{r_i}/d + m_i$, which implies $m_i = \hat{r_i}/d + n_i$, this is the expected value:

$$\begin{aligned}
&((v_i^{\hat{r_i}})^{-t_i/d} v_i^{t_i m_i}(g^x)^{dm_i}, (v_i^{\hat{r_i}})^{(-1/d)} v_i^{m_i}) \\
&= ((v_i^{\hat{r_i}})^{-t_i/d} v_i^{t_i(\hat{r_i}/d+n_i)}(g^x)^{d(\hat{r_i}/d+n_i)}, (v_i^{\hat{r_i}})^{(-1/d)} v_i^{(\hat{r_i}/d)+n_i)}) \\
&= (v_i^{t_i n_i}(g^x)^{\hat{r_i}+dn_i}, v_i^{n_i}) = (g^{x\hat{r_i}}(v_i^{t_i} g^{xd})^{n_i}, v_i^{n_i}) \\
&= (g^{x\hat{r_i}}(v_i^{t_i} g^{x(\mathsf{ID}-\mathsf{ID}^*)})^{n_i}, v_i^{n_i}) = (g^{x\hat{r_i}}(w_i g^{x\mathsf{ID}})^{n_i}, v_i^{n_i})
\end{aligned}$$

The second term is uniformly distributed among all elements in $\mathbb{Z}_p^*$ due to the selection of m_i. Private keys can be generated for all identities except ID^*.

Challenge(m_0, m_1) : $\mathcal{B}$ picks random bit $b \in \{0,1\}$. The response is: $(C_0$, $(C_{A,1}$, $C_{B,1})$, ... ,$(C_{A,k}$, $C_{B,k}))$. $\mathcal{B}$ sets $C_0 = m_b T$ and for each i from 1 to k it sets:

$$C_{A,i} = (g^y)^{a_i}, \quad C_{B,i} = (g^y)^{a_i \cdot t_i}.$$

We observe that $(g^y)^{a_i} = v_i^{y_i}$ and that $(g^y)^{a_i t_i} = v_i^{t_i y_i} = (w_1 u^{\mathsf{ID}^*})^{y_i}$ from which correctness follows. The simulator's ability to construct the second term in this manner follows directly from the fact that the encrypted identity is ID^* and no g^x term appears in $w_1 u^{\mathsf{ID}^*}$.

For each value of i, this implicitly sets $ya_i = s_i y_i$ or $y_i = ya_i/s_i$. If the input is a valid k-BDH tuple then the response is drawn from a uniform distribution and $m_b T$ is the expected value:

$$\begin{aligned}
m_b T &= m_b \prod_{1 \le i \le k} e(g,g)^{xyr_i} = m_b \prod_{1 \le i \le k} e(g^x, g^{s_i r_i/a_i})^{ya_i/s_i} \\
&= m_b \prod_{1 \le i \le k} e(g^x, v_i^{r_i/a_i})^{y_i} = m_b \prod_{1 \le i \le k} e(g^x, v_i^{\hat{r_i}})^{y_i}
\end{aligned}$$

If T is not a valid k-BDH tuple then the distribution is uniform and independent of b.

Phase 2: $\mathcal{A}$ issues more private key queries. It is exactly the same as Phase 1.

Guess: $\mathcal{A}$ outputs a guess of $b' \in \{0, 1\}$. If $b = b'$ then $\mathcal{B}$ outputs 1 meaning T is a valid k-BDH tuple. Otherwise, it is not a valid k-BDH tuple and the output is 0.

When the input is a valid k-BDH instance, $\mathcal{A}$ must satisfy $|\Pr[b = b'] - \frac{1}{2}| \geq \epsilon$. When the input is not a valid k-BDH instance, the input is uniform and independent and $\Pr[b = b'] = \frac{1}{2}$. Therefore, we have, as required:

$$|\Pr[\mathcal{B}(\text{valid } k\text{-BDH}) = 1] - \Pr[\mathcal{B}(\text{not valid } k\text{-BDH}) = 1]| \geq |(\frac{1}{2} + \epsilon) - \frac{1}{2}| \geq \epsilon.$$

3.2 Efficiency

Assume that the value $e(g^x, v_i^{r_i})$ is precomputed for all values $1 \leq i \leq k$. Each encryption takes k exponentiations and k group operations in $\mathbb{G}_T$, $2k + 1$ exponentiations and k group operation in $\mathbb{G}$. Decryption requires $2k$ bilinear map computations, one inversion and $2k + 2$ group operations in $\mathbb{G}_T$.

3.3 Extensions

This construction fits in Boneh-Boyen framework. We give the natural extension to a hierarchical IBE and to a fully secure IND-CPA scheme in the style of [26] in the full version of the paper [23].

4 Relationship between *k*-BDH Problems

In this section we prove that the k-BDH family of problems becomes progressively weaker. Informally, this means that an oracle for k-BDH does not help in solving a $(k + 1)$-BDH instance.

The proof uses the generic group model [35,20,21]. This idealized version of a group retains the important properties of the group while facilitating reasoning about its minimal possible assumptions. If a statement cannot be proven in the generic group model then it is impossible to find a group for which the statement holds. The generic group model has been used to reason about complexity assumptions both with bilinear maps [5,6] and without bilinear maps [21].

The closely related proof for the separation of k-Linear family of assumptions [10] could not be used directly. This stems from the fact that a standard multilinear map [36] cannot be used to solve k-BDH. We create a modified k-multilinear map that takes as input k elements in $\mathbb{G}$ and 1 element in $\mathbb{G}_T$ (which is the result of a bilinear map on two elements in $\mathbb{G}$) and produces an output in a third group $\mathbb{G}_M$ (the target group of the k-multilinear map). The modified k-multilinear map acts as an oracle for k-BDH. The main technical difficulty is showing that all inputs to the k-multilinear map fail to produce a multiple of the target element in the $(k + 1)$-BDH instance.

Theorem 6. *If the k-BDH assumption holds, then so does the $(k+1)$-BDH assumption.*

Proof. Informally, this means that if $(k+1)$-BDH is easy, then k-BDH is also easy. Suppose we have an oracle $\mathcal{A}$ for $(k+1)$-BDH. $\mathcal{A}$ can be used to solve an k-BDH instance $(g,\, g^x,\, g^y,\, v_1,\, \ldots,\, v_k,\, v_1^{r_1},\, \ldots,\, v_k^{r_k},\, T)$. Select random $v_{k+1} \in \mathbb{G}$ and $r_{k+1} \in \mathbb{Z}_p^*$ and run $\mathcal{A}$ on input $(g,\, g^x,\, g^y,\, v_1,\, \ldots,\, v_k,\, v_{k+1},\, v_1^{r_1},\, \ldots,\, v_k^{r_k},\, v_{k+1}^{r_{k+1}},\, T \cdot e(g^x, g^y)^{r_{k+1}})$. By returning the same value as $\mathcal{A}$, the simulation is perfect.

As in the k-Linear generic group separation proof [10], we prove a stronger result by means of a multilinear map [36,37] in Theorem 7. A k multilinear map is an efficiently computable map $e_k : \mathbb{G}^k \to \mathbb{G}_M$ such that $e_k(g_1^{a_1}, \ldots, g_k^{a_k}) = e_k(g_1, \ldots, g_k)^{\prod_{i=1}^k a_i}$ for all $g_1, \ldots, g_k \in \mathbb{G}$ and $a_1, \ldots, a_k \in \mathbb{Z}_p$; and $e_k(g, \ldots, g) \neq 1$. Here, we consider a modified k-multilinear map: $\hat{e_k} : \mathbb{G}_T \times \mathbb{G}^k \to \mathbb{G}_M$ where $\mathbb{G}_T$ is the group resulting from a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We define $\hat{e_k} : (e(g_x, g_y)^{a_T}, g_1^{a_1}, \ldots, g_k^{a_k}) = \hat{e_k}(e(g_x, g_y), g_1, \ldots, g_k)^{a_T \prod_{i=1}^k a_i}$.

Lemma 1. *Given a modified k-multilinear map there is an efficient algorithm to solve k-BDH.*

Proof. On input a k-BDH instance $(g, g^x, g^y, v_1, \ldots, v_k, v_1^{r_1}, \ldots, v_k^{r_k}, T)$ output "yes" if $\hat{e_k}(T, v_1, \ldots, v_k) \stackrel{?}{=} \prod_{i=1}^{k} \hat{e_k}(e(g^x, g^y), v_1, \ldots, v_{i-1}, v_i^{r_i}, v_{i+1}, v_k)$ and "no" otherwise. This is correct because

$$\prod_{i=1}^{k} \hat{e_k}(e(g^x, g^y), v_1, \ldots, v_{i-1}, v_i^{r_i}, v_{i+1}, v_k) = \prod_{i=1}^{k} \hat{e_k}(e(g,g), v_1, \ldots, v_k)^{xyr_i}$$
$$= \hat{e_k}(e(g,g), v_1, \ldots, v_k)^{xy \sum_{i=1}^k r_i}$$

and when $T = e(g,g)^{xy \sum_{i=1}^k r_i}$ equality holds as required.

In the generic group model, elements of $\mathbb{G}$, $\mathbb{G}_T$ and $\mathbb{G}_M$ are encoded as opaque strings such that only equality can be tested by the adversary. To perform operations in the group the adversary queries oracles. The oracles map the opaque string representations to elements of $\mathbb{G}$, $\mathbb{G}_T$ and $\mathbb{G}_M$ using ξ_G,ξ_T and ξ_M respectively. In our case, we provide the adversary with oracles to perform Group Action in each group, Inversion in each group, Bilinear Map for $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and Modified k-Multilinear Map for $\mathbb{G}_T \times \mathbb{G}^k$.

Theorem 7. *Let $\mathcal{A}$ be an algorithm that solves $(k+1)$-BDH in the generic group model making a total of q queries to the oracles computing the group action in $\mathbb{G}$, $\mathbb{G}_T$ and $\mathbb{G}_M$, the oracles computing inversion in $\mathbb{G}$, $\mathbb{G}_T$ and $\mathbb{G}_M$, the bilinear map oracle and an oracle for modified k-multilinear map. Then $\mathcal{A}$'s probability of success is bounded by*

$$\epsilon \leq \frac{(k+5)(q+2k+5)^2}{p}.$$

Proof. Consider an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$ as follows.

Let g be a randomly selected generator of $\mathbb{G}$. Select random x, y, v_1, ..., v_{k+1}, r_1, ..., r_{k+1}, $c \in \mathbb{Z}_p$ as well as random bit $d \in \{0,1\}$. Set $T_d = e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$ and $T_{1-d} = e(g,g)^c$. $\mathcal{A}$ is given ($\xi_G(g)$, $\xi_G(g^x)$, $\xi_G(g^y)$, $\xi_G(g^{v_1})$, ..., $\xi_G(g^{v_{k+1}})$, $\xi_G(g^{v_1 r_1})$, ..., $\xi_G(g^{v_{k+1} r_{k+1}})$, $\xi_T(T_0)$, $\xi_T(T_1)$) with the goal of guessing d.

$\mathcal{B}$ keeps track of the elements known to $\mathcal{A}$ as three lists: $L_G = \{(F_{G,i}, \xi_{G,i})\}$, $L_T = \{(F_{T,i}, \xi_{T,i})\}$ and $L_M = \{(F_{M,i}, \xi_{M,i})\}$. The first element of each list is the internal representation kept by $\mathcal{B}$- represented as a polynomial in the ring $\mathbb{Z}_p[1, x, y, v_1, \ldots, v_{k+1}, r_1, \ldots, r_{k+1}, c]$. The set of all elements in these rings are denoted F_G, F_T and F_M. The second element is the opaque representation known to $\mathcal{A}$. $\mathcal{B}$ handles oracle queries from $\mathcal{A}$ by calculating the correct value and checking to see if a the corresponding external representation already exists. If so, the corresponding known representation is returned; otherwise $\mathcal{B}$ generates a distinct random string to serve as the external representation and adds it to the respective list. We assume that the domains of ξ_G, ξ_T and ξ_M are sufficiently large so that the probability that algorithm $\mathcal{A}$ makes queries for an element other than one obtained through $\mathcal{B}$ is negligible.

Oracle queries from $\mathcal{A}$ are handled by $\mathcal{B}$ as follows:

Group Action: Given elements in $\mathbb{G}$ with internal representations $F_{G,i}$ and $F_{G,j}$ compute $F' = F_{G,i} + F_{G,j}$. If there does not already exist an external representation of the value F' then generate $\xi_G(F')$ and add $(F', \xi_G(F'))$ to L_G . Return $\xi_G(F')$. Group Action for $\mathbb{G}_T$ and $\mathbb{G}_M$ are handled analogously. Denote the number of Group Action queries made in $\mathbb{G}$ as q_{G_g}, the number of Group Action queries made in $\mathbb{G}_T$ as q_{T_g} and the number of Group Action queries made in $\mathbb{G}_M$ as q_{M_g}.

Inversion: Given an element in $\mathbb{G}$ with internal representation $F_{G,i}$ set $F' = -F_{G,i}$. If there does not already exist an external representation of the value F' generate $\xi_G(F')$ and add $(F', \xi_G(F'))$ to L_G . Return $\xi_G(F')$. Inversion for $\mathbb{G}_T$ and $\mathbb{G}_M$ are handled analogously. Denote the number of Group Action queries made in $\mathbb{G}$ as q_{G_i}, the number of Group Action queries made in $\mathbb{G}_T$ as q_{T_i} and the number of Group Action queries made in $\mathbb{G}_M$ as q_{M_i}.

Bilinear Map (e): Given elements in $\mathbb{G}$ with internal representations $F_{G,i}$ and $F_{G,j}$ calculate $F' = F_{G,i} \cdot F_{G,j}$. If there does not already exist an external representation of the value F' generate $\xi_T(F')$ and add $(F', \xi_T(F'))$ to L_T. Return $\xi_T(F')$. Let q_B denote the number of bilinear map queries made.

Modified k-Multilinear Map ($\hat{e_k}$): Given elements in $\mathbb{G}$ with internal representations $F_{G,v1}$, ..., $F_{G,vk}$, and an element in $\mathbb{G}_T$ with internal representation $F_{T,j}$. Compute $F' = F_{T,j} \prod_{i=1}^{k} F_{G,vi}$. If there does not already exist an external representation of the value F' generate $\xi_M(F')$ and add $(F', \xi_M(F'))$ to L_M. Return $\xi_M(F')$.

Elements in F_G have at most degree 2; elements of F_T have at most degree 4; elements in F_M have degree at most $2k + 4$. The input elements that are in $\mathbb{G}$ have corresponding elements in F_G with degree at most 2 and the elements in $\mathbb{G}_T$ have corresponding elements in F_T with degree at most 3. The group action and inversion operations cannot increase the degree of the polynomials in F_G,F_T

or F_M. The Bilinear Map operation uses elements in $\mathbb{G}$ to produce elements of at most degree $2 + 2 = 4$ in F_T. The Modified Multilinear Map produces elements of at most degree $4 + k(2)$ in F_M.

Finally, $\mathcal{A}$ halts and outputs a guess of d' for d. $\mathcal{B}$ now selects random $g^* \in \mathbb{G}$ and x^*, y^*, v_1^*, …, v_{k+1}^*, r_1^*, …, r_{k+1}^*, $c^* \in \mathbb{Z}_p$. T_b is set to $e(g^*, g^*)^{x^* y^* \sum_{i=1}^{k+1} r_i^*}$ and $T_{1-b} = e(g^*, g^*)^{c^*}$. All elements besides T_b are independent of each other. Therefore the simulation engineered by $\mathcal{B}$ is consistent with these values unless one of the following events occur:

- Two values in F_G have the same representation in $\mathbb{G}$
- Two values in F_T have the same representation in $\mathbb{G}_T$
- Two values in F_M have the same representation in $\mathbb{G}_M$
- Using a bilinear map on values in F_G, it is possible to find a multiple of $e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$ in F_T.
- Using a modified k-multilinear map on values in F_G, it is possible to find a multiple of $e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$ in F_M.

The input elements are all chosen independently. Since $\mathcal{A}$ makes $q_{G_g} + q_{G_i}$ group actions or inversion queries for group $\mathbb{G}$ the corresponding elements in F_G are at most degree 2 and the probability of a collision is $\binom{q_{G_g}+q_{G_i}+2(k+1)+3}{2}\frac{2}{p}$. For the elements in $\mathbb{G}_T$ there $q_{T_g} + q_{T_i} + q_B$ group actions or inversion or bilinear map queries are made resulting in elements in $\mathbb{G}_T$. Since elements in $\mathbb{G}_T$ have corresponding polynomials in F_T with degree at most 4 the probability of a collision is $\binom{q_{T_g}+q_{T_i}+q_B+2}{2}\frac{4}{p}$. For each of the group actions in $\mathbb{G}_M$, inversion in $\mathbb{G}_M$ and k-Modified Multilinear Map queries the probability of a collision is $\binom{q_{M_g}+q_{M_i}+q_K}{2}\frac{2k+4}{p}$.

Next, we show the probability of finding a multiple of $e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$ from the terms in F_G is zero. If a multiple exists, it must be formed using at least one bilinear map operation. Since x, y, r_i all appear in T_b then the product of at least two of these values must appear in the same element in F_G for each value of i, $1 \leq i \leq k + 1$. This is impossible by the following claim:

Claim: It is impossible for any two of x, y, r_i to appear in the same monomial in F_G:

Proof. We show that each way to choose two of the three values to appear in the same term is impossible:

- x and y appear in the same term. This requires creating a multiple of the polynomial xy. We are initially given the polynomials x and y each of degree 1 (and polynomials that are independent of x and y). This means from polynomial of degree 1 we must create a polynomial of degree 2 also in F_G. Only the group action and inversion oracles result in new elements in F_G. However, the output of these oracles cannot increase the degree of a monomial. Thus we cannot create monomials of degree greater than 1 from x and y. In particular, xy cannot be created by the adversary.

- x and r_i appear in the same term. This requires creating a multiple of xr_i namely axr_i. Since the term r_i never appears without v_i it follows that $v_i \mid a$ and we can rewrite axr_i as $a'xr_iv_i$. This is a polynomial of degree 3. It is impossible to create a polynomial of degree greater than 2 in F_G. So x and r_i cannot appear in the same term.
- y and r_i appear in the same term. This follows from a symmetric argument that x and r_i cannot appear in the same term.

Finally, we claim that it is impossible to find a multiple of $e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$ in F_M. In order to use the modified k-multilinear map to find a multiple of a target value, $T_d \stackrel{?}{=} e(g^x, g^y)^{\sum_{i=1}^{k+1} r_i}$, at least one $\hat{e_k}$ operation involving a multiple of T_d is required. The only option is to use T_d as the input element in $\mathbb{G}_T$. The modified k-multilinear map produces a multiple of $xy \sum_{i=1}^{k+1} r_i$, namely $Axy \sum_{i=1}^{k+1} r_i$. $\mathcal{A}$ must then use combination of oracle calls, denoted $\mathcal{F}$, using only values in F_G to form $Axy \sum_{i=1}^{k+1} r_i$ so that it can test equality.[3]

All inputs in F_G containing r_i also contain v_i. As a result, any monomial divisible by r_i is also divisible by v_i. Every type of oracle call preserves this property. In particular, consider the polynomial $Axy \sum_{i=1}^{k+1} r_i$ constructed by the adversary in $\mathcal{F}$. It is required that each monomial in the expansion of $Axyr_i$ must be divisible by v_i. It follows that for each of the $k+1$ values of v_i it is the case that $v_i | A$.[4] Specifically, A is divisible by $\prod_{i=1}^{k+1} v_i$.

For a given value i, the value $Axyr_i$ is divisible by $k+4$ values: x, y, v_1, …, v_{k+1}, and r_i. Producing such a term requires taking the product of at least $k+3$ terms available to the adversary (x and y only appear on their own and it is impossible to produce a multiple of v_iv_j in F_G using the group action and inversion oracles). However, the bilinear map can only take the product of 2 values and the modified k-multilinear map can only take the product from a bilinear map and k additional values for a total of $k+2$. Consequently, we deduce that the adversary cannot synthesize a multiple of $xy \sum_{i=1}^{k+1} r_i$ in F_M to cause a collision.

The probability of finding a collision is bounded by

$$\begin{aligned}\epsilon &\leq \binom{q_{G_g} + q_{G_i} + 2(k+1) + 3}{2} \frac{2}{p} + \binom{q_{T_g} + q_{T_i} + q_B + 2}{2} \frac{4}{p} \\ &\quad + \binom{q_{T_m} + q_{T_i} + q_k}{2} \frac{(2k+4)}{p} \\ &< \frac{(q+2k+5)^2 + 2(q+2)^2 + (k+2)q^2}{p} < \frac{(k+5)(q+2k+5)^2}{p}\end{aligned}$$

The combination of these two theorems implies: DBDH=1-BDH $\lneq$ 2-BDH $\lneq$ $\cdots \lneq$ k-BDH $\lneq$ $(k+1)$-BDH $\lneq \cdots$.

[3] $\mathcal{A}$ could first perform $\hat{e_k}(CT_d + D, n_1, \ldots, n_k)$ for constant C and a polynomial D that does not contain T_d. It would then perform some combination of oracle calls,$\mathcal{F}$, to produce a value equal to $\hat{e_k}(CT_d + D, n_1, \ldots, n_k)$. However, an equivalent test is to first perform $\hat{e_k}(CT_d, n_1, \ldots, n_k)$ and then test equality with $\mathcal{F}/\hat{e_k}(D, n_1, \ldots, n_k)$.

[4] For a more detailed argument see [10].

4.1 Relationship between k and the Group Size

From Theorem 7, we know that increasing k increases security. The generic attack on k-BDH appears to require $O(k)$ discrete logarithm calculations, and that solving t discrete logarithm problems on a given curve appears to require $O(t)$ times solving one problem; assuming that, the generic attack scales linearly with k.

Another means of increasing security is to increase the group size. An interesting question is, "what is the equivalent increase in group size if we increase k to $k+1$." We assume finding the discrete log is a function, f, of the order of the group. Then in the generic attack on k-BDH where $\mathbb{G}$ has prime order p increasing k to $k+1$ is approximately equivalent to increasing the group size from p to $f^{-1}(\frac{(k+1)f(p)}{k})$.

5 Conclusions and Future Work

We have proposed k-BDH, a family of assumptions generalizing the DBDH assumption. We have given evidence, using the generic group model, that assumptions in the k-BDH family become strictly weaker with increasing values of the parameter k. Unlike the k-Linear family of assumptions, k-BDH makes a natural tool for constructing pairing-based cryptosystems, including IBEs. We have demonstrated this by constructing a family of IBEs in which the kth member is secure based on k-BDH. Our IBE family fits in the Boneh-Boyen framework. Our k-BDH family allows IBEs to be instantiated with an assumption safety buffer for the first time.

We hope that, like k-Linear, our k-BDH assumption family will see widespread use. We believe that it will be especially well suited for constructing attribute-based encryption and other forms of functional encryption. In addition, we believe that dual system encryption techniques could be applied to k-BDH, yielding more efficient cryptosystems with tighter security reductions.

An important open problem arises from the fact that the k-BDH assumptions are all no weaker than computational BDH (just as the k-Linear assumptions are all no weaker than CDH). Because the components of our IBE grow with k, there may be a crossover point beyond which an IBE based on hard-core bits of the computational BDH problem is more efficient than one based on k-BDH.

References

1. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. Computing 32(3), 586–615 (2003); Extended abstract in Proceedings of Crypto 2001
2. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)

3. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
4. Abdalla, M., Pointcheval, D.: Interactive Diffie-Hellman Assumptions with Applications to Password-Based Authentication. In: S. Patrick, A., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 341–356. Springer, Heidelberg (2005)
5. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
6. Boyen, X.: The Uber-Assumption Family – A Unified Complexity Framework for Bilinear Groups. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 39–56. Springer, Heidelberg (2008)
7. Gentry, C., Halevi, S.: Hierarchical Identity Based Encryption with Polynomially Many Levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)
8. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
9. Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
10. Shacham, H.: A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074 (2007), `http://eprint.iacr.org/`
11. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
12. Lewko, A.B., Waters, B.: Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In: ACM Conference on Computer and Communications Security, pp. 112–120. ACM (November 2009)
13. Boneh, D., Montgomery, H., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: Keromytis, A., Shmatikov, V. (eds.) Proceedings of CCS 2010, pp. 131–140. ACM Press (October 2010)
14. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
15. Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
16. Naor, M., Segev, G.: Public-Key Cryptosystems Resilient to Key Leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
17. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: Trevisan, L. (ed.) Proceedings of FOCS 2010, pp. 511–520. IEEE Computer Society (October 2010)
18. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)

19. Boyle, E., Segev, G., Wichs, D.: Fully Leakage-Resilient Signatures. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 89–108. Springer, Heidelberg (2011)
20. Nechaev, V.I.: Complexity of a determinate algorithm for the discrete logarithm. Mathematical Notes 55(2), 165–172 (1994)
21. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
22. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
23. Benson, K., Shacham, H., Waters, B.: The k-bdh assumption family: Bilinear map cryptography from progressively weaker assumptions. Cryptology ePrint Archive, Report 2012 (2012), http://eprint.iacr.org/
24. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)
25. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Atluri, V., Meadows, C., Juels, A. (eds.) Proceedings of CCS 2005, pp. 320–329. ACM Press (November 2005)
26. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
27. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of ACM Conference on Computer and Communications Security 2006, pp. 89–98. ACM (November 2006)
28. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
29. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Dwork, C. (ed.) Proceedings of STOC 2008, pp. 187–196. ACM (May 2008)
30. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
31. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of STOC 1989, pp. 25–32. ACM (1989)
32. Verheul, E.R.: Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 195–210. Springer, Heidelberg (2001)
33. Moody, D.: The diffie hellman problem and generalization of verheuls theorem. Designs, Codes and Cryptography 52, 381–390 (2009)
34. Canetti, R., Halevi, S., Katz, J.: A Forward-secure Public-key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
35. Babai, L., Szemerédi, E.: On the complexity of matrix group problems I. In: Proceedings of FOCS 1984, pp. 229–240. IEEE Computer Society (October 1984)
36. Boneh, D., Silverberg, A.: Applications of multilinear forms to cryptography. In: Topics in Algebraic and Noncommutative Geometry: Proceedings in Memory of Ruth Michler. Contemporary Mathematics, vol. 324, pp. 71–90. American Mathematical Society (2003)
37. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices and applications. Cryptology ePrint Archive, Report 2012/610 (2012)